Residual Attention Network for Image Classification

E4040.2020Fall.YEAH.report
Jinzhu Yang jy3024, Jiayun Ni jn2722, Saier Gong sg3772
Columbia University

Abstract

In this work, we do a comprehensive review of the paper 'Residual Attention Network for Image Classification", reproduce the results with Tensorflow and Python, and compare between the obtained results and the results in the original paper. With the same architecture, we build up Attention-56 and Attention-92 models containing residual units and attention modules. We train the models on CIFAR-10, CIFAR-100 and tiny ImageNet datasets, with the same methods of data augmentation, optimization and testing. Also, several comparisons are done to validate the effectiveness of each component in Residual Attention Network as well as the robustness against noisy labels.

During the process, we have met some problems such as unclarified hyperparameter settinging in the paper, limited computational resources and training time. Therefore, we refer to other papers, combining fine tuning to achieve the reasonable parameter settings for our models, such as the kernel size, strides, the number of epochs, etc. But, this may cause our different results from the original paper. The test error rate of our model is around 18%. We get lower accuracy than the result of around 5% in the original model. In addition, when comparing the influence of different attention types in Attention-56, we get the best result with Spatial Attention, while in the original paper, Mixed Attention is the best. However, in most comparisons, we obtain the same conclusions as the original paper, proving the better performance of certain structures.

1. Introduction

Image classification has become a fundamental task to be addressed in many fields, including self-driving cars, and facial recognition. Numerous images are produced everyday, providing us with convenience and confusion. Learning correctly from images is thus quite crucial. Computers could now perform better than or comparable to human by deep networks using convolutional layers in several image recognition tasks. Intuitively, computers would improve its accuracy by mimicking the procedure of visual recognition from human. Inspired by the selective attention to parts of a scene for quick perception by human visual system, for example focusing on the road instead of the sky while we are driving, attention mechanism in deep learning has improved the success of various models in recent years. Attention mechanism first emerged in natural language processing. Later, this mechanism was used in various other applications, including computer vision, speech processing, etc. [1]

During the past several decades, methodologies and techniques[2] of image classification have revolved rapidly and many great accomplishments have been achieved by researchers all over the world. Convolutional neural networks(CNN), which helps to extract important information from images, are widely used to deal with image classification problems. However, deep neural networks would result in the problem of exploding gradient or vanishing gradient. Such problem has been alleviated through the use of normalized initialization[3] and intermediate normalization layers[3]. ResNet model, one type of deep residual learning model[4], exists later with an identity part between layers also helps to solve the vanishing gradient problem. In image processing, the issue with such tasks is that there is often a complicated dependency along the spatial domain. To be specific, multiple disjoint semantic cues in multiple areas of an image may give clues to the overall representation of an image and this dependency may vary case by case. As a result, there emerges a need to learn a flexible dependency mechanism that, given an input image, the machine can figure out which parts are most important to overall representation. [5] For this challenge, attention, a mechanism by which a network can weigh features by level of importance to a task, and use this weighting to help achieve the task, seems an attractive solution. Based on the above thinking, many forms of attention have been developed, such as global and local attention, soft and hard attention, self-attention, multi-head attention, etc. [1, 5, 6] Visual attention mechanism [7] has thus been utilized for image captioning with several different attention types.

The authors of the paper Residual Attention Network for Image Classification[8] added attention module to residual learning model to improve model performance. This creative combination achieves great performance for image classification on the CIFAR-10, CIFAR-100 and ImageNet datasets with error rates of 3.9%, 20.45% and 4.8% respectively in the paper. In this project, we aim to reconstruct the Residual Attention Network (Attention-56 and Attention-92) and Naive Attention Model (NAL) and reproduce those error rates by training them with CIFAR-10, CIFAR-100 and ImageNet datasets.

The background and aim of this project are introduced in this section, followed by the summary of the methodology and fundamental idea in the referenced paper [8] in section 2. In section 3, the methodology of our project is stated. Section 4 and section 5 demonstrate

the detailed implementation of our models including the model architecture and hyper-parameters. We show the results and comparison to the original paper. Conclusion and discussion are also derived accordingly. Acknowledgements of our work from others and some references of papers and researches are listed in section 6 and 7. Section 8 is the attached appendix, providing the individual contributions in our group.

2. Summary of the Original Paper

2.1 Methodology of the Original Paper

In the referenced paper [8], authors of the paper proposed "Residual Attention Network", a convolutional neural network using residual learning with attention mechanism, which can incorporate with the state-of-art feed forward network architecture in an end-to-end training fashion. An example architecture of the proposed network is shown in Figure 2.1.

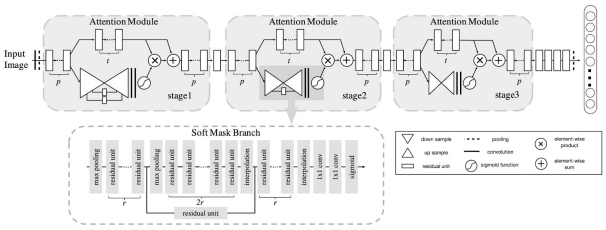

Figure 2.1: Example architecture of the proposed network and the structure for soft mask branch in stage 2.

Mask branch and trunk branch

They constructed the residual attention network by stacking multiple attention modules, each divided into mask branch and trunk branch Figure 2.2. The trunk branch would perform the work of feature processing while the mask branch would serve as feature selector. The mask branch is devised as a symmetrical bottom-up top-down architecture, which fulfills the bottom-up fast feedforward process and top-down attention feedback in one single feedforward process. Max pooling is performed as down sampling and bilinear interpolation is performed as up sampling. They also add skip connections between bottom-up and top-down parts to capture information from different scales.

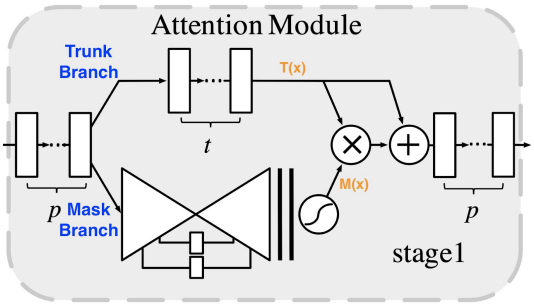

Figure 2.2: Attention module architecture for stage1 in residual attention network.

Attention Residual Learning

Instead of naively stacking the attention modules together, which might cause poor performance, the authors introduced the idea of attention residual learning, which proposed that the soft mask branch would be constructed as an identical mapping part of the attention module before final output. The output of the Attention Module using attention residual learning would be modified as

$$H_{i,c}(x) = (1 + M_{i,c}(x)) * F_{i,c}(x)$$

where i ranges over all spatial positions and c ranges over the index of channels. $M_{i,c}(x)$ represents the output of mask branch and $F_{i,c}(x)$ indicates the features generated from the trunk branch as shown in Figure 2.2.

Under such formulation, attention residual learning would keep the good properties of original features but also pass them directly into top layers to weaken the feature selection ability of mask branches.

The detailed values of the parameters in Residual Attention Network for CIFAR-10, CIFAR-100 and ImageNet would be a little bit different due to the fundamental discrepancies in the image data. An example of the detailed architecture of the Residual Attention Network for ImageNet is shown in Table 4.2.

Experiments

Authors of the original paper did a series of contrast experiments on CIFAR-10, CIFAR-100 and ImageNet datasets to evaluate the performance of Residual Attention Network from two aspects: (1) The effectiveness of the components in Residual Attention Network including attention residual learning mechanism and different architectures of soft mask branch in the Attention Module. (2) The noise resistance property of Residual Attention Network.

2.2 Key Results of the Original Paper

The Effectiveness of the Residual Learning

As demonstrated in the paper, the Attention-56 model and Attention-92 model using residual attention network

would achieve Top-1 error rate of 5.52% and 4.99% respectively, while obtaining Top-1 error rate of 5.89% and 5.35% with naive attention learning algorithm, which proves the effectiveness of ARL model.

Network	ARL(Top-1 err. %)	NAL(Top-1 err. %)
Attention-56	5.52	5.89
Attention-92	4.99	5.35

Table 2.1: Classification error (%) on CIFAR-10.

Different Attention Types

Attention type has also been stated to be influential to the Top-1 error rate. Three types of activation functions listed below are used, corresponding to mixed attention, channel attention and spatial attention. Mixed attention outperforms the others with Top-1 error rate of 5.52%.

Activation Function	Attention Type	Top-1 err. (%)
$f_1(x)$	Mixed Attention	5.52
$f_2(x)$	Channel Attention	6.24
$f_3(x)$	Spatial Attention	6.33

Table 2.2: Test error (%) on CIFAR-10 of Attention-56 Network with different attention types.

Different Mask Structures

Different mask structures are also considered in the experiments to validate the effectiveness of encoder-decoder structure by comparing with local concolutions without any down sampling or up sampling. They used the Attention-56 model to construct both. After training two models on CIFAR-10, they found that encoder-decoder structure has better performance, with Top-1 error rate of 5.52%.

Mask Type	Attention Type	Top-1 err.(%)
Local Concolutions	Local Attention	6.48
Encoder-Decoder	Mixed Attention	5.52

Table 2.3: Test error (%) on CIFAR-10 using different mask structures.

Noisy Label Robustness

Experiment on the noise resistant property for the residual attention network has also been done. They explored the performance of Attention-92 model and ResNet-164 model with noise levels of 10%, 30%, 50% and 70% and the Top-1 error rates of them are stated below. The test error of Attention-92 network is smaller than that of ResNet-164 network with same noise level. Also, Attention-92 network obtained test error decreasing slowly than ResNet-164 network when increasing noise level, showing the noise-resisting property of RAL model.

Noise Level	ResNet-164 err. (%)	Attention-92 err. (%)
10%	5.93	5.15
20%	6.61	5.79
50%	8.35	7.27
70%	17.21	15.75

Table 2.4: Test error (%) on CIFAR-10 with label noises.

Attention-92 model was applied to both datasets of CIFAR-10 and CIFAR-100 and gained Top-1 error rates of 4.99% and 21.71% respectively.

3. Methodology (of the Students' Project)

We applied the same model structure, tried our best to keep the settings like hyperparameters and dataset the same as the paper and did some deduction to the settings not specified in this paper with the references. In addition, considering the limited computation resources, we do fine tuning of the model parameters, skip the extreme deep network (Attention-128, Attention-164), use the results of other state-of-the-art methods like ResNet-152, WRN-16-8, etc. as reference, and use alternative dataset tiny ImageNet 200. (ImageNet we mention in the following parts indicates tiny ImageNet 200.)

3.1. Objectives and Technical Challenges

Our objective is to reproduce all the experiment results given by Attention Residual Network and compare our results with the results in the original paper.

The challenge is that due to the unclear description of certain parameter settings and the limited computation resources, it is hard for us to do a completely identical implementation, such as the same training epochs, number of filters and kernel size in convolutional layers.

3.2.Problem Formulation and Design Description

1. **Validation the effectiveness of the residual learning.** We compare the top-1 error of attention residual learning (ARL) and 'naive attention learning' (NAL) with Attention-56 and Attention-92 respectively on CIFAR-10 dataset.

2. **Comparison of mixed attention f_1, channel attention f_2 and spatial attention f_3.** Constraints to attention have been added to mask branch by changing normalization step in activation function before soft max output. We compare the top-1 error of three types of attention by using different activation functions. We do this experiment on CIFAR-10 dataset with Attention-56 with different activation functions.
$f_1 = \frac{1}{1+exp(-x_{i,c})}$
$f_2 = \frac{x_{i,c}}{||x_i||}$
$f_3 = \frac{1}{1+exp(-(x_{i,c}-mean_c)/std_c)}$

3. **Comparison of different mask structures.** We compare the top-1 error of encoder-decoder structure in Attention Residual Network with local convolutions without any down sampling or up sampling on CIFAR-10. We directly use the result of the local attention model in the paper to compare with our gained Attention-56 result.

4. **Noisy Label Robustness.** We compare ResNet-164 network with Attention-92 network under different noise levels. We directly use the result of ResNet-164 network from the paper to compare with our gained Attention-92 network.

$$Q = \begin{pmatrix} r & \frac{1-r}{9} & \cdots & \frac{1-r}{9} \\ \frac{1-r}{9} & r & \cdots & \frac{1-r}{9} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-r}{9} & \frac{1-r}{9} & \cdots & r \end{pmatrix}_{10\times 10}$$

Figure 3.1 : The confusion matrix Q, where r is the clean label ratio for the whole dataset.

Following paper [9], we add label flip noise with label distribution by confusion matrix Q. $q_{ij} = p(y = j|y^* = i)$, y^* denotes the true labels with 10 classes. y denotes the noisy labels. r denotes the clean label ratio for the whole dataset.

5. **State-of-art methods comparison.** We compare our Residual Attention Network (Attention-56 and Attention-92) with other state-of-the-art methods including ResNet, Wide ResNet and deeper Attention models on the aspects of parameter size, Top-1 error rate on CIFAR-10 and CIFAR-100, respectively. We directly use the result of other state-of-the -art methods from the paper.

4. Implementation

In this section, we provide detailed description of the model structure in section 4.1, show the data description, data augmentation, optimization and test method in section 4.2.

4.1. Deep Learning Network

4.1.1 Residual Unit

The authors of the original paper do not specify more detailed structure of the residual units used on CIFAR except that they use pre-activation structure and keep most of the settings same as RestNet paper [10]. In RestNet paper, the authors use non-bottleneck design residual units with zero-padding identity shortcuts in all cases. We finally choose two kinds of bottleneck design residual units, with both projection shortcut and identity shortcut, because of the following reasons: (1) Bottleneck design can reduce the number of the parameters in the model and decrease the training time. (2) The zero-padded dimensions in identity shortcuts have no residual learning. [10] We decide to use these two kinds of Residual Units for both CIFAR and ImagNet models. The structure is illustrated in Figure 4.1, 4.2.

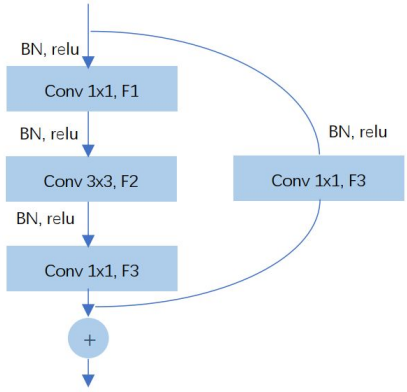

Figure 4.1 : Pre-activation Residual Unit with projection shortcut (Res_conv)

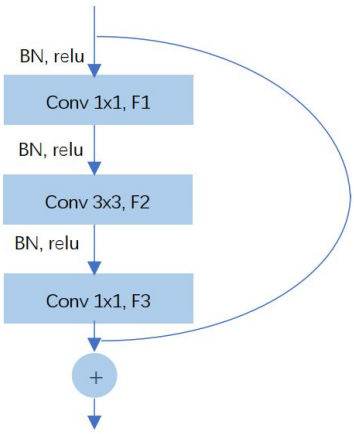

Figure 4.2 : Pre-activation Residual Unit with identity shortcut (Res_identity)

4.1.2 Attention Block

The structure for attention blocks for both CIFAR and ImageNet models are the same. There are three attention blocks with different structures corresponding to three attention stages as Figure 2.1 from the original paper. They use the hyper-parameters setting: {p = 1, t = 2, r = 1}, demonstrate the whole architecture of three attention stages and the detailed structure of the soft mask branch in stage 2. However, the structure info for the soft mask branch in stage 1 and 3 is unclear in the paper. Figure 2.1 only shows the detailed structure with one skip connection of stage 2, from which we can just estimate the approximate structure of stage 1 and stage 3: For stage 3, there is no skip connection, and for stage 1, there are two skip connections. But we are unclear about the exact structure like the number of max pooling and interpolation layers of stage 1 and 3. Therefore, we check the code containing only the models for ImageNet on github [11] published by one of the authors of the original paper, and ascertain the structure for the other two stages as Figure 4.3, 4.4. In addition, we add batch normalization and ReLu activation layer after each interpolation and the first 1x1 convolutional layer in the mask branch which are omitted in Figure 2.1, 4.3, 4.4.

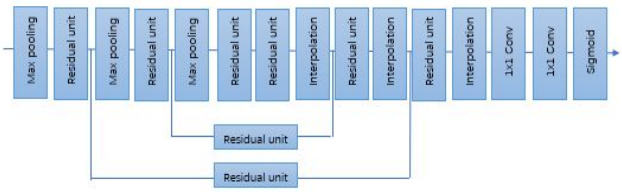

Figure 4.3 : The structure of the soft mask branch for stage 1.

There are three pairs of down sampling and up sampling structure. Besides, there are two skip connections between bottom-up and top-down parts.

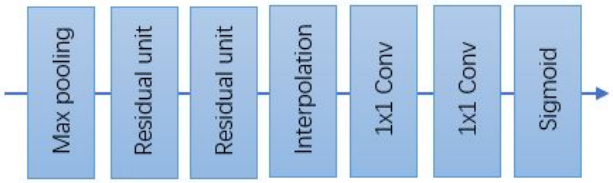

Figure 4.4 : The structure of the soft mask branch for stage 3.

There is only one pair of down sampling and up sampling structure and no skip connections.

4.1.3 Residual Attention Network

We find there are slight differences between the whole architecture in Figure 2.1 and Table 2 in the paper. There are four Residual Units at the top part of the network in Figure 2.1 but three in Table 2. After checking the official code on Github [11], we use the architecture with three Residual Units at the top part of the network.

Models for CIFAR

The authors of the original paper do not provide the hyper parameters for CIFAR (CIFAR-10, CIFAR-100) as what they provide for ImageNet. Since they claim that ResNet is used as their baseline method. We constructed our own Attention-56 and Attention-92 models referring to the settings in ResNet paper [10], especially for the number of filters, kernel size, strides in convolutional layers. The subsampling is performed by convolutions with a stride of 2. And if the feature map size is halved, the number of filters is doubled so as to preserve the information as much as we can. The input size of CIFAR is 32x32x3 and the architecture is as shown in Table 4.1. We adopt the weight initialization method following previous study [12], where the authors derive a theoretically more sound initialization for extremely deep models. In each layer containing parameters, we first count the total number of weight parameters ($n_l = num_{filter} * kerne_{height} * kernel_{width}$), then we sample these parameters from a zero-mean Gaussian distribution whose standard deviation is $\sqrt{\frac{2}{n_l}}$. We also initialize the bias parameters equal to 0.

Layer	Output Size	Attention-56	Attention-92
Conv1	32x32x16	3x3, 16	
Max pooling	32x32x16	3x3	
BN+ReLU	32x32x16	/	
Res_identity	32x32x16	1x1, 4 3x3, 4 1x1, 16	

Attention Stage 1	32x32x16	x1	x1
Res_conv	16x16x32	1x1, 8 3x3, 8, stride 2 1x1, 32	
Attention Stage 2	16x16x32	x1	x2
Res_conv	8x8x64	1x1, 16 3x3, 16, stride 2 1x1, 64	
Attention Stage 3	8x8x64	x1	x3
Res_identity x3	8x8x64	1x1, 16 3x3, 16 1x1, 64	
BN+ReLU	8x8x64	/	
Global average pooling	1x1x64	/	
FC,Softmax	10 for CIFAR-10, 100 for CIFAR-100		

Table 4.1 : Residual Attention Network architecture details for CIFAR-10 and CIFAR-100. The unspecified strides are all 1 as default value.

Models for ImageNet

The authors of the original paper clearly show the architecture details for ImageNet. However, due to large parameter size, we meet out of memory error in the training. In addition, we use the tiny ImageNet 200 dataset instead which has different input size 64x64x3 and smaller number of classes. We fine tune the hyper parameters and construct our own Attention-56 network described in Table 4.2. The main differences lie in the number of filters for convolutional layers and considering the smaller input size, we use stride 1 instead of 2 for the first convolutional layer, so that the output size of the top layers will not be too small due to multiple down sampling processes. We initialized the weights in each layer from a zero-mean Gaussian distribution with standard deviation 0.01 and the neuron biases with constant 0. [13]

Layer	Output Size for our Attention-56	Our Attention-56	Original Attention-56
Conv1	64x64x16	3x3x64	7x7, 64, stride 2
Max pooling	32x32x16	3x3, stride2	3x3, stride 2
BN+ReLU	32x32x16	/	
Res_identity	32x32x16	1x1, 16 3x3, 16 1x1, 64	1x1, 64 3x3, 64 1x1, 256
Attention Stage 1	32x32x16	x1	x1
Res_conv	16x16x128	1x1, 32 3x3, 32,stride 2 1x1, 128	1x1, 128 3x3, 128,stride 2 1x1, 512
Attention Stage 2	16x16x128	x1	x1
Res_conv	8x8x256	1x1, 64 3x3, 64,stride 2 1x1, 256	1x1, 256 3x3, 256,stride 2 1x1, 1024
Attention Stage 3	8x8x256	x1	x1
Res_identity x3	4x4x512	1x1, 128 3x3, 128,stride 2 1x1, 512	1x1, 512 3x3, 512,stride 2 1x1, 2048
BN+ReLU	4x4x512	/	
Global average pooling	1x1x512	/	
FC,ReLU	1024	/	
FC,Softmax	200	/	

Table 4.2 : Residual Attention Network architecture details for ImageNet and the comparison with the model in original paper.

4.2. Experiment Detail

According to the implementation methods of the original paper, we use different data augmentation and optimization methods for CIFAR and ImageNet datasets separately.

4.2.1 Data Description

CIFAR

CIFAR-10 and CIFAR-100 datasets consist of 60,000 32x32 color images of 10 and 100 classes respectively

which was splitted into 40,000 training images, 10,000 validation images and 10,000 test images.

ImageNet

ImageNet LSVRC 2012 dataset [14] used in the original paper contains the 1000 categories and 1.2 million images. The validation and test data will consist of 150,000 photographs. We use tiny ImageNet 200 [15] to replace LSVRC 2012 due to limited computational resources and training time we can afford. Tiny ImageNet 200 contains 200 classes. The images are uniformly distributed in all the classes. All images are 64x64 colored ones. We splitted the data into 40,000 training images, 10,000 validation images and 2,000 test images.

4.2.2 Preprocess and Data Augmentation

CIFAR

The preprocess and data augmentation method for datasets of CIFAR-10 and CIFAR-100 are rather straightforward as mentioned in the paper. We use the same preprocessing and augmentation methods on CIFAR-10 and CIFAR-100 datasets. Each image is padded by 4 pixels on each side, filled with 0 resulting 40x40 image size. Then 32x32 crop is randomly sampled from an image and its horizontal flip, with the per-pixel RGB mean value subtracted.

ImageNet

For preprocessing, we scale the pixel value to [0,1], with mean value subtracted and standard variance divided. Since the training set is too large and loading all the training set will cause out of memory error. We randomly sample 10,000 images from the training set which are used to calculate the mean and standard deviation.

For data augmentation, we follow the practice in previous study [13]. We apply scale and aspect ratio augmentation [16]: First, we randomly choose the percentage of the area of the patch to the original image, which is evenly distributed between 8% and 100%. Then, the aspect ratio is chosen randomly between 3/4 and 4/3. Furthermore, we implement standard color augmentation by performing PCA on the set of RGB pixel values throughout the above sample of training set, getting three eigenvalues λ_1, λ_2, λ_3 and three eigenvectors p_1, p_2, p_3 . For each training image, we draw one random variable from a Gaussian distribution with mean 0 and standard deviation 0.1 until this image is used for training again. Then, to each RGB pixel $I_{xy} = [\, {I_{xy}}^R + {I_{xy}}^G + {I_{xy}}^B]^T$, we add the following quantity: $[p_1, p_2, p_3]\cdot[\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$.

4.2.3 Optimization

CIFAR

We use the same optimizer in both CIFAR-10 and CIFAR-100 datasets. The network is trained using nesterov SGD with a mini-batch size of 64, a momentum of 0.9. Together, we set the initial learning rate to 0.1 and a decay of 0.0001. The learning rate is divided by 10 at 100 and 150 iterations. Unlike the 160k iterations in the original paper, we only train 180 epochs with an early stopping method with patience equals to 10.

ImageNet

The network is trained using SGD with a momentum of 0.9, initial learning rate to 0.001. Unlike the 530k iterations in the original paper, we train 45 epochs with an early stopping method with patience equals to 3.

4.2.4 Test Method

CIFAR

The data of original 32x32 images are used to evaluate the performance of the networks built for CIFAR-10 and CIFAR-100.

ImageNet

We use 10 crop testing to calculate the both top-1 error and top-5 error. The network makes a prediction by extracting five 56×56 patches (the four corner patches and the center patch) as well as their horizontal reflections (hence ten patches in all), resizing them to 64x64, and then averaging the predictions made by the network's softmax layer on the ten patches. [13]

5. Results

5.1. Project Results

CIFAR

After implementation, we obtained the following results. Values of top-1 error rates of each model and their training time could be summarized in these five tables below. Some of the figures demonstrating the training loss and accuracy are shown in this section. You could see all the similar output figures of each model trained in our github repository.

The Attention-56 and Attention-92 models using ARL network are established and are trained and evaluated by CIFAR-10 and CIFAR-100 datasets with results shown in table 5.1 and accuracy plots in figure 5.1. The Attention-92 model outperforms the Attention-56 model on CIFAR-10 with test error rate of 16.21% compared with 16.59%. But the situation is inverse for the CIFAR-100 dataset.

Network	CIFAR-10	CIFAR-100

	(Training Time)	(Training Time)
Attention-56	16.59 (2h25min8s)	47.20 (3h18min)
Attention-92	16.21 (2h 40min 24s)	49.87 (1h 57min 50s)

Table 5.1 : Test error (%) on CIFAR-10 and CIFAR-100

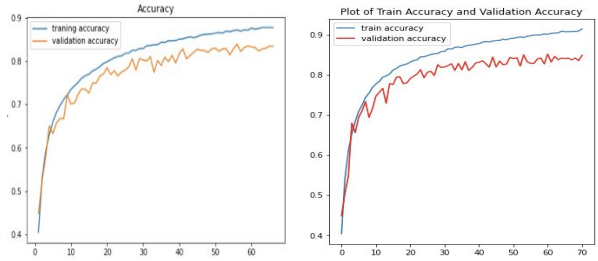

Figure 5.1 : Accuracy plot for Attention-56 and Attention-92 on CIFAR-10.

We compare the performance of Attention-56 and Attention-92 model built using ARL network and NAL network. The results show that ARL network (figure 5.1) outperforms NAL network (figure 5.2) with test error rates of both attention models built by ARL network smaller than that of models built by NAL network, as shown in table 5.2. This is consistent with the original paper, proving the effectiveness of ARL network we built.

Network	ARL(Top-1 err. %) (Training Time)	NAL(Top-1 err. %) (Training Time)
Attention-56	16.59 (2h25min8s)	17.95 (2h59min58s)
Attention-92	16.21 (2h 40min 24s)	25.57 (2h 26min 3s)

Table 5.2 : Test error (%) on CIFAR-10 of ARL and NAL.

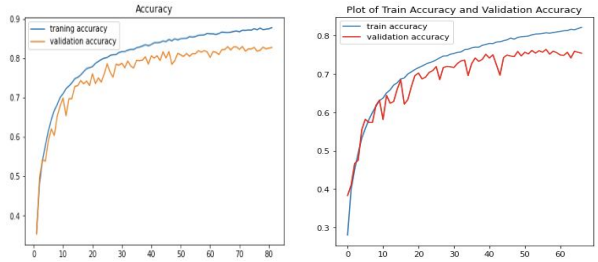

Figure 5.2 : Accuracy plot for Attention-56 and Attention-92 using NAL on CIFAR-10.

Different attention types are also implemented in Attention-56 model in this project. As shown in table 5.3, the test error rates of the three attention types are 16.59%, 19.16% and 15.36% respectively, with Attention-56 model using spatial attention gaining the highest accuracy.

Activation Function	Attention Type	Top-1 err. (%) (Training Time)
$f_1(x)$	Mixed Attention	16.59 (2h25min8s)
$f_2(x)$	Channel Attention	19.16 (1h56min53s)
$f_3(x)$	Spatial Attention	15.36 (4h43min24s)

Table 5.3 Test error (%) on CIFAR-10 of Attention-56 network with different attention types.

The performance of the Attention-92 model with different level of label noises have been tested as well. Table 5.4 lists the top-1 error rate and training time for each level of noises. Attention-92 model with 50% noises would be thought to be converging the fastest since the training time of it is the shortest. Attention-92 model with 10% label noises is the best of the four with top-1 error rate of 20.64%, but outperformed by the Attention-92 model without noises. The accuracy of the model would slightly decrease with increasing level of noises, proving the noise-resisting property of residual attention network.

Noise Level	Attention-92 err. (%)	Training Time
10%	20.64	2h 40min 55s
30%	24.99	1h 50min 55s
50%	38.74	1h 20min 20s
70%	54.83	1h 51min 42s

Table 5.4: Test error (%) on CIFAR-10 with label noises.

ImageNet

Table 5.5 shows the result on ImageNet of Attention-56. The model performs similarly on training and validation set, but better on test set. Top-1 error is largely higher than top-5 error.

Network	Training accuracy	Validation accuracy	Test Top-1 error	Test Top-5 error	Training Time /Epoch
Attention-56	0.3614	0.3605	0.5875	0.019	486s 345ms

Table 5.5 : Train, validation and test error on ImageNet

5.2. Comparison of the Results Between the Original Paper and Students' Project

For CIFAR part, we get lower accuracy than the original paper in each model due to our hyperparameter tuning such as epochs to save training time. However, when comparing the performance between ARL and NAL, we get the same conclusion as the original paper, that ARL is better than NAL. What' more, when comparing the performance between Attention-56 and Attention-92 on CIFAR-10 dataset, Attention-92 ARL is better than Attention-56, same as the original paper, but Attention-56 NAL has better performance than Attention-92, which is different from the original paper. Moreover, when training on CIFAR-100 dataset, Attention-56 is better than Attention-92, but worse than the same model and other state-of-the-art methods in the paper. To test the noise robustness of ARL, we experimented on the CIFAR-10 dataset using Attention-92 with noise level of 10%, 30%, 50% and 70% respectively. The results show the same trend of obtaining less accuracy with higher level of noise introduced into data as the original paper. When only considering Attention-56, we finish three types of model with mixed, channel and spatial attention, respectively, and find out that the spatial attention has the best performance, which is different from the paper where mixed attention is the best.

For the ImageNet part, we get lower accuracy than original paper and our model seems less fitting due to our hyper parameter tuning to save the computation resources. However, although we turn to another alternative smaller dataset, the Residual Attention Network still takes effect, especially considering the top-5 error which is even lower than the result with LSVRC 2012 in the original paper. Moreover, our similar results for the training, validation set and even better result on the test set indicates the data augmentation method used in the paper is effective in avoiding overfitting.

5.3. Discussion of Insights Gained

For CIFAR part, several reasons may cause the lower accuracy of each model than those stated in the original paper. Because of time constraints, we reduce our training epoch to 180 with early stopping where patience is 10, which will make the training process stop early. Then, when considering the parameters for model, like kernel size, the number of filters, we do deduction from the baseline model in the original paper which may cause different results.

For ImageNet, higher top-1 error can be induced by the following reasons. Due to limited computational resources, we adjust the hyper-parameters like a smaller number of filters and kernel size in convolutional layers and decrease the number of downsampling processes by using stride 1 in the first convolutional layer alternatively.

Although we comply with the rule of thumb that we add more filters as the size of the feature map decreases, the smaller number of the filters may cause the model less fitting. Since each filter performs a different convolution on the input to the layer and extract different information, we may miss some information and get lower classification accuracy.

The smaller kernel size and the smaller stride in pooling layers may also harm the performance of our model. They decrease the receptive field which is an indication of the scope of input data a neuron within a layer can be exposed to.

And the fewer downsampling processes may do damage to generalization. The downsampling process in convolutional layer or pooling layer has a strong motivation to remove sensitivity to small translations of the input images.

We also turn to smaller datasets. Smaller datasets may cause overfitting, but since we use data augmentation methods and decrease the model complexity, obvious overfitting does not appear in our experiment.

6. Conclusion

In this work, we reproduce the paper "Residual Attention Network for Image Classification". We build our own Residual Attention Networks including Attention-56 and Attention-92 with fine tuning and the similar implementation, each of which is made up of our self defined residual units and Attention Modules. Following the ideas of the original paper, we run our models on CIFAR-10, CIFAR-100 and tiny ImageNet datasets, do some comparison and obtain the following results:

- The accuracy of each model is smaller than that in the original paper, probably due to the hyperparameter tuning, for example epochs.
- When training on CIFAR-10 dataset, Attention-92 is better than Attention-56, but on CIFAR-100 dataset, Attention-52 is slightly better than Attention-96, which is different from the original paper.
- When training on CIFAR-10 dataset, Attention-56 is better than NAL-56, and Attention-92 is better than NAL-92.

- When doing the comparison of different mask structures, we get that Spatial Attention has the best performance, while in the paper, Mixed Attention is the best.
- When testing the effect of different noisy labels on Attention-92 model, lower noise performs better.
- Residual Attention Network also performs well on tiny ImageNet dataset.

During the process, we have learned by ourselves a complicated model, from data preprocessing to test error analyses, from hyperparameter tuning to model training. In order to completely reproduce the paper, we have read a lot of papers and codes, and had many discussions, during which we gradually optimize these models. Besides the model architecture itself, we also understand the mathematics behind each layer and parameter tuning step.

As we stated, our training result is different from the original paper, therefore, for future research, we would like to do more research about the hyperparameter setting such as kernel size and number of filters. Also we want to try more ways to avoid overfitting except early stopping such as L2-norm. Moreover, if there is enough time, we would like to train more epochs to see if we can get a better result.

6. Acknowledgement

During the project, besides some related papers and codings as we mentioned in section 7 References, our course slides also help a lot. Since we need to be familiar with the basic ResNet, our slides provide us with a complete explanation about that. What's more, Piazza is a good place to solve common questions such as the difficulty in software and so on, it saves us much time. Moreover, at the beginning of this project, we were upset of the model implementation since it seemed a little bit complicated for us, however, thanks to Professor's comfort and encouragement, we have relaxed a lot and quickly clarified our goal.

7. References

[1] American Express, "Attention mechanism in deep learning," Analyticsvidhya.com, 20-Nov-2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/11/comprehensive-guide-attention-mechanism-deep-learning/. [Accessed: 19-Dec-2020].

[2] Siddhartha Sankar Nath, Jajnyaseni Kar, Girish Mishra, Sayan Chakraborty, "A Survey of Image Classification Methods and Techiniques", 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014.

[3] Manishgupta, "Understanding ResNet and its Variants", Towards Data Science, 2020

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", ArXiv (cs.CV), 2015.

[5] S. Sohail, "A survey of visual attention mechanisms in deep learning," Medium, 05-Dec-2019. [Online]. Available: https://medium.com/@shairozsohail/a-survey-of-visual-attention-mechanisms-in-deep-learning-1043eb25f343. [Accessed: 19-Dec-2020].

[6] A. Lihala, "Attention and its different forms - towards data science," Towards Data Science, 29-Mar-2019. [Online]. Available: https://towardsdatascience.com/attention-and-its-different-forms-7fc3674d14dc. [Accessed: 19-Dec-2020].

[7] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, Tat-Seng Chua, "SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning", ArXiv (cs.CV), 2016.

[8] F. Wang et al., "Residual Attention Network for Image Classification," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[9] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," arXiv [cs.CV], 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," arXiv [cs.CV], 2015.

[11] Official code: https://github.com/fwang91/residual-attention-network

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", ArXiv (cs.CV), 2015.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012, vol. 25, pp. 1097–1105.

[14] "ImageNet large scale visual recognition competition 2012 (ILSVRC2012)," Image-net.org. [Online]. Available: http://image-net.org/challenges/LSVRC/2012/. [Accessed: 20-Dec-2020].

[15] http://cs231n.stanford.edu/tiny-imagenet-200.zip

[16] C. Szegedy et al., "Going deeper with convolutions," arXiv [cs.CV], 2014.

8. Appendix

8.1 Individual Student Contributions in Fractions

	UNI1	UNI2	UNI3
Last	jy3024	jn2722	sg3772

Name	Jinzhu Yang	Jiayun Ni	Saier Gong
Fraction of (useful) total contribution	1/3	1/3	1/3
What I did 1	Coding for CIFAR and ImageNet model (residual unit, attention model, three attention types, robustness test), load data and preprocessing, data augmentation (standard color aug, scale and aspect ratio aug) and testing	Coded up function of crop-testing, the module for NAL, ARL92 models, NAL56 model and NAL92 model	Coded for CIFAR, including the layer structures (residual units, attention modules) and data preprocessing for CIFAR; build up the model ARL56.
What I did 2	Trained ImageNet model	Trained models of NAL92(CIFAR-10), ARL92(CIFAR-10, CIFAR-100), ARL92_noise10(CIFAR-10),ARL92_noise30(CIFAR-10), ARL92_noise50(CIFAR-10),ARL92_noise70(CIFAR-10)	Trained models of NAL56(CIFAR-10), ARL56(CIFAR-10, CIFAR-100), ARL56 with channel attention(CIFAR-10), ARL56 with spatial attention(CIFAR-10).
What I did 3	Report	Report	Report